An Interop Publication

ConneXions®

The Interoperability Report

January 1996 Volume 10, No. 1

ConneXions—
The Interoperability Report tracks current and emerging standards and technologies within the computer and communications industry.

In this issue:

ConneXions is published monthly by Interop Company, a division of SOFTBANK Exposition and Conference Company, 303 Vintage Park Drive, Foster City, California, 94404–1138, USA.
Phone: +1 (415) 578-6900
Fax: +1 (415) 525-0194
E-mail: **connexions@interop.com**

Subscription hotline: 1-800-575-5717
or +1 610-892-1959

ISSN 0894–5926

From the Editor

Happy New Year and welcome to Volume 10 of *ConneXions—The Interoperability Report*. We begin 1996 with a second article adapted from the book *Building Internet Firewalls* by Brent Chapman and Elizabeth Zwicky. (The first article appeared in the December 1995 issue.) This month's article is a comprehensive explanation of how to develop Internet Security Policies for your site. Securing your network goes far beyond technical solutions such as firewalls, you'll also need procedures for dealing with everything from user passwords to your legal responsibilities with respect to corporate shareholders.

Over the last few months I have been experimenting with various kinds of digital imaging and audio on my workstation. This is pretty straight forward, and provides me with a great excuse to have some fun while learning about new technologies. Most of today's computers allow you to connect a digital still or video camera to the appropriate port and capture images which can be manipulated with a variety of software packages. Alternatively, you may wish to download pictures or movie clips from public repositories and view them using simple, often "freeware," applications. If you did not know it already, you soon discover that movie clips and high-quality still images take up a lot of storage space, require fast processors to manipulate and take a long time to download. But these limitations pale in comparison to what you have to endure if you attempt to send digital audio and video *live* over the network. It is generally agreed that modern packet switching networks were not designed with this kind of traffic in mind, but that's not to say it cannot be done. In our series *Back to Basics,* François Flückiger discusses the fundamental requirements for sending audio and video over computer networks.

While preparing the audio/video article, I decided to expand the acronym "SECAM" and add it to the glossary for completeness. Using the "Net Search" feature of Netscape, I had no problem locating a number of sources for this information; the one I ended up using is operated by the BBC in Aberdeen, Scotland. The Web has indeed become a useful information tool, perhaps a little too useful as indicated by the first hit I got when searching for SECAM: "Welcome to SECAM. Stanford Engineering Club for Automation and Manufacturing. SECAM offers leadership opportunities to MSE students." :–)

We've managed to keep our subscription prices unchanged for the last five years, but increasing costs—particularly a sharp rise in the cost of paper—have made a moderate price adjustment necessary at this time. The good news is that existing subscribers will be offered a renewal discount, and the price of back issues will remain at $15. The 1995 index sheet is currently being printed and will be mailed to you with the February 1996 issue. As always, you can download the latest index information from **http://www.interop.com.**

Internet Security Policies

by
Brent Chapman, Great Circle Associates
and
Elizabeth Zwicky, Silicon Graphics

Introduction

The word "policy" makes many people flinch, because it suggests impenetrable documents put together by unknowledgeable committees, which are then promptly ignored by everyone involved (except when they make a good excuse or weapon). That's not the kind of policy we're discussing in this article.

The policy we're talking about here is like a nation's foreign policy. It might be discussed in documents—of varying amounts of legibility—but its primary purpose is to lay out a direction, a theory of what you're trying to achieve. People sometimes confuse the words "policy," "strategy," and "tactics." A *policy* is what determines what wars you're going to fight and why. A *strategy* is the plan for carrying out the war. A *tactic* is a method for carrying out a strategy. Presidents determine policy; generals determine strategies; and anybody down to a foot soldier might determine a tactic.

Most of our book [from which this article is adapted] is about tactics. The tactics involved in building a firewall, the nitty-gritty details of what needs to be done here, are complex and intricate. However, no matter how good your tactics are, if your strategy and policy are bad, you can't succeed. In the 1800s, an American named William Walker set out to conquer Nicaragua for the United States. His strategy and tactics were, if not impeccable, certainly successful: he conquered Nicaragua. Unfortunately, there was a literal fatal flaw in his plan. The United States did not at the time want Nicaragua, and when he announced that he had conquered it, the U.S. government was completely uninterested in doing anything about it. Walker ended up ruling Nicaragua very briefly, before he was killed in a popular uprising. This was the result of getting the strategy and the tactics right, but completely botching the policy.

Your Security Policy

Most technical computer people consider a single, unified, published security policy to be desirable in the abstract, but believe—with a strong basis in personal experience—that attempting to come up with one is going to be extremely painful. For example, walk up to any system administrator and ask about users and passwords, and you are almost guaranteed to be rewarded with a rant. Everybody has a story about the apparent insanity of people faced with passwords, one of the simplest and most comprehensible security issues: the professor who explained that he was too important to need a good password; the mathematician who was told that he couldn't use a password because it was in an English dictionary (and who replied that he wasn't using the *English* word that was spelled that way, he was using the *Russian* word that was spelled that way, and nobody had told him not to use Russian words). This kind of experience is apt to convince system administrators that their user community is incapable of dealing intelligently with security issues.

There is no doubt that putting together a security policy is going to be a long, involved process, and that it's the exact opposite of the types of tasks most technical people enjoy. If you like to program, you are extremely unlikely to enjoy either the meetings or the bureaucracy involved in policy making. On the other hand, putting together a security policy is a great deal more amusing than dealing with the side effects of not having one. In the long run, you'll spend less time in meetings arguing about security if you get it out of the way ahead of time.

Developing a security policy also doesn't need to be as bad as you may be expecting. Many of the problems with security policies are caused by people who are trying to write a security policy that sounds like a security policy, which is to say that it's written in big legal and technical words and says threatening things about how users had better behave themselves. This doesn't work. It's also the most unpleasant way to do things, because it involves hostility and incomprehension all around. It's true that your organization may at some point need a security policy that's written in big legal words (to satisfy some big legal requirements). In that case, the security policy you write shouldn't contradict the legalistic document, but the policy you write doesn't need to be that legalistic one.

Another problem people have in trying to write security policies is that they have a strong feeling about what the policy ought to be, and they're uncomfortable that the actual policy they enforce does not meet that standard. There is a great deal of lip service paid to the notion that security should be absolute: you should have a site that nobody could ever break in to; where every user has exactly one account, and every account has exactly one user; and where all the passwords are excellent, and nobody ever uses anybody else's password for anything.

In the real world, nobody's site is like that, a fact that is well-known and well-accepted. That doesn't keep people from claiming that they want to make their site like that, sometimes in big words on many pieces of paper that they call a security policy. Invariably, every time, without exception, these policies are not followed by anybody.

It's unlikely that your policy is one that emphasizes security at all costs. Such a policy would be irrational. It is reasonable to value other things highly enough to be willing to compromise security.

Most houses would be more secure with bars over all the windows. Few people are willing to put bars over their windows, despite a desire to protect themselves. People have a number of reasons for compromising their security in this way. To start with, bars are expensive and they interfere with using the windows for many of their normal purposes (e.g., seeing out of, climbing out of in an emergency). But people are willing to go to equal expense and inconvenience to apply other security solutions, and they may avoid barring windows even when it's the cheapest and most convenient solution, because it looks bad and makes them feel oppressed.

This is entirely reasonable, and it's entirely reasonable to make the same type of decision about your computer security. You may not want the best security money can buy, or even the best security you can afford.

Requirements

What do you want? You want the best security that meets your requirements for:

- *Affordability:* How much money does the security cost?
- *Functionality:* Can you still use your computers?
- *Cultural compatibility:* Does it conflict with the way people at your site normally interact with each other and the outside world?
- *Legality:* Does it meet the your site's legal requirements?

continued on next page

Internet Security Policies *(continued)*

Don't pretend that you want to be absolutely secure, if only you could afford it. You don't live your life with the most perfect security money could buy. For the same reasons, it's extremely unlikely that your institution can maintain the characteristics that are important to it if it also installs the most perfect security money could buy. People don't like learning or working in a hostile environment; because they won't do it, you'll either lose the security or lose the organization.

Sometimes a small concession to insecurity can buy a large payoff in morale. For example, rulemakers reel at the idea of guest accounts, but a guest account for a spouse can make a big difference in how people feel about work. And there are sometimes unexpected results. One university computer center was asked why its student employees were allowed to hang around at all hours, even after the labs were closed, doing random activities of dubious value to the computer center; it seemed insecure at best. The answer was that several years before, an operator who was typing his girlfriend's term paper in a lab after hours had discovered and responded to a critical emergency. Because he had saved the facility from what seemed likely to be a million dollars worth of uninsured damage (insurance companies have a nasty tendency to consider floods in windowless third-floor computer rooms to be acts of God, and thus uninsurable), the computer facility management figured that all the computer time the operators wanted had already been paid for.

On the other hand, if you have too little security, you can lose the organization to lawyers or attackers, and what matters there is what you do, not what you write down. Writing down marvelous policies that don't get enforced certainly won't save you from people who are trying to break into your computer, and it generally won't save you from lawsuits, either. The law counts only policies that you make some attempt to enforce. Writing it down and brazenly not doing it proves that you aren't simply too stupid to know what to do: it demonstrates that you actually knew what you had to do, and didn't do it!

What should a Security Policy contain?

First and foremost, a security policy is a way of communicating with users and managers. It should tell them what they need to know to make the decisions they need to make about security.

Explanations

It's important that the policy be explicit and understandable about why certain decisions have been made. Most people will not follow rules unless they understand why they're important. A policy that specifies what's supposed to be done, but not why, is doomed. As soon as the people who wrote it leave, or forget why they made those decisions, it's going to stop having any effect.

Everybody's responsibilities

A policy sets explicit expectations and responsibilities among you, your users, and your management; it lets all of you know what to expect from each other. It's a mistake to distribute a policy that concentrates entirely on what users need to do to make the site secure (it seems hostile and unfair), or entirely on what system administrators need to do (it encourages the users to believe that somebody else will handle it, and they don't have to worry about it).

Regular language

Most people are not lawyers, and they're not security experts. They're comfortable with casual descriptions. You may be afraid to write a policy that way because it may seem uncomfortably casual and too personal. But it's more important to make your security policy friendly and understandable than to make it precise and official-looking.

Write it as if you were explaining it to a reasonably bright but non-technical friend. Keep it a communication between peers, not a memo from Mount Olympus. If that's not acceptable in your corporate culture, write two separate policy descriptions.

You will not get people to comply unless they understand the document and want to comply with it, and that means they have to at least be willing to read it. If they shut their brains off in paragraph two because the document sounds legal and threatening, you lose. You also lose if they decide that you think they're stupid, or if they decide that you don't care. Don't get so informal that you seem condescending or sloppy. If necessary, get a technical writer to clean up the punctuation and spelling.

Enforcement authority

Writing down the policy is not the point; living by it is. That means that when the policy isn't followed, something should happen to fix it. Somebody needs to be responsible for making those corrections happen, and the policy needs to specify who that's going to be and the general range of corrections. Here are some examples of what a security policy might specify:

- Managers of certain services have the authority to revoke access.
- Managers will be asked to take care of some kinds of transgressions.
- Facilities that don't meet certain standards may be cut off from the corporate network and external access by the people who run the corporate network.

The policy should specify who is going to decide and give some indication of what kinds of penalties are available to them. It should not specify exactly what will happen when; it's a policy, not a mandatory sentencing law.

Provision for reviews

You can't expect to set a policy up once and forget it. The needs of your site will change over time, and policies that were perfectly sensible may become either too restrictive or too lax. Sometimes change is obvious: if you work for a startup company that goes from six people to 6,000 people, it will probably occur to you that things are different in important ways (but you still may not get around to redoing the security policy if you didn't set up a mechanism for that in advance). If you work for a 200-year old university, however, you may not expect much change. However, even if the organization appears to be doing its best to fossilize, the computers change, the external networks change, and new people come in to replace ones who leave. You still need to review and change your policies on a regular basis.

Discussion of specific security issues

Because of the differences between organizations, it's hard to be specific about issues without writing an entire book just about security policies. However, here are some common issues to consider when you are writing a policy:

- Who is allowed to have an account at your site? Do you have guest accounts? What do you do about contractors, vendors, and clients?
- Can accounts be shared between multiple people? What about a secretary who uses an executive's account to process that person's electronic mail? What about joint projects? What about family members? Is it sharing an account if you let somebody else borrow a window on your machine really quickly?

continued on next page

Internet Security Policies *(continued)*

- When do people lose the privilege of having an account, and what do you do about it? What happens if people leave or are denied access?
- Who can set up dial-in modems? Is it OK for other people to set up dial-out modems? Is there anything special about PPP, SLIP, or ISDN lines?
- What do people need to do before they connect a computer to the main network?
- How secure do computers need to be before they get services from centrally maintained machines?
- How secure do computers need to be in order to connect to a network with unprotected access to the Internet?
- How is financial data going to be protected?
- How is confidential information about people going to be protected?
- What do individual users need to do to protect themselves and the site? What kinds of passwords should they have, and when should they change them?
- What can people do on the Internet? Should they be transferring random executables in and running them?
- What precautions do you need to take against viruses on personal computers?
- Who can connect your site to external networks, and what's an external network? Is it OK for a project manager to connect your site to another specific site? How about putting in a second Internet connection?
- How are home computers going to be secured? How are they going to get secure access to your network?
- How are people who are traveling going to get access to the network?
- What information is considered company confidential? How is it going to be protected? Can it be sent outside the site via electronic mail?
- If you have remote sites, how are they going to get secure access to your main network?

What should a Security Policy not contain?

Some pieces of information don't belong in your site's security policy, as we discuss in this section.

Technical details: The security policy needs to describe what you're trying to protect and why; it doesn't necessarily need to describe the details of how. It's much more useful to have a one-page document that describes "what" and "why" in terms that everyone in your organization can understand, than a 100-page document that describes "how," but that nobody except your most senior technical staff can understand.

For example, consider a policy that includes a requirement that says:

> "Nonreusable passwords shall be used to authenticate all incoming connections from the outside world, in order to prevent potential attackers from being able to capture reusable passwords by monitoring such connections."

This requirement is much more useful than a policy that says:

"S/Key will be used for all incoming connections."

Why? Because the first policy describes *what* is to be protected and *why,* and it leaves *how* open so the technical staff can select the best implementation.

A policy that says the following is better yet:

"Regular passwords are often stolen and reused when they pass across networks. We won't use passwords that can be reused across networks our company doesn't control."

This policy communicates the same information without the legal-style language. It also clarifies some other points. For example, in the original language does the "outside world" include companies that have special relationships with yours? It may seem obvious to you that it does, but it probably doesn't seem obvious to the managers who are arranging to work with those companies. The reworded language makes it clear what the criterion is (although you may still end up arguing about what networks meet it).

Policy can guide you in selecting and implementing technology, but it shouldn't be used to specify it. It's often much easier to get management to buy into, and sign off on, an overall policy than on a specific technology.

Somebody else's problem: Every site's security policy is different. Different sites have different concerns, different constraints, different users, and different capabilities; all of these lead to different policies. Further, a site's policy may change over time, as the site grows and changes. Don't assume that you need to do things the way they've always been done, or that you can borrow somebody else's policy and simply change the names in it.

Putting together a Security Policy

Once you know what you want in a security policy, how do you put one together? The first step towards putting together a working security policy for your site is to decide what your personal opinion is. If you've been administering a site or making any decisions about security, you've been enforcing an internal theory about security, even if you've never articulated it. You're going to need to come to a clear and explicit understanding of what that internal policy is before you can discuss policy issues with other people in order to produce a written policy for your site.

With that in mind, look at the decisions you've made about security and decide what you think your site's security goals should be. That may not be the policy that your site ends up with, but it's an important first step.

What is your site's Security Policy?

The second step towards putting together a working security policy for your site is to determine what everybody else's security policy is. What do the users and managers expect security to do for them? What do they think of the way security is handled currently? What are other computer facilities doing and why?

Every site has at least one security policy. The problem is that most sites have more than one; perhaps as many as there are people involved with the site's computers. Sometimes this proliferation of policies is purely unconscious; different computer facilities within the same site may be doing radically different things without even realizing it.

continued on next page

Internet Security Policies *(continued)*

Sometimes it's an open secret; administrators may be trying to maintain a security policy that they believe is necessary, even though the user population does not agree with them. Sometimes it's out-and-out war. Generally, people think of universities as the main place where computer users and computer administrators are engaged in open security warfare, but in fact many companies spend large amounts of time fighting about security issues (for example, administration and the engineers are often at odds).

Some of the security policies for a site may be written down already, but most are likely to be implicit and unpublicized. The only way to find out about them is to go and ask. Be sure to ask managers, system administrators, and users. Then look at the actual computers and see what's really going on. It's unlikely that anybody will actually lie to you. However, they may be telling you what they think is going on, or what they wish was going on, or what they know is supposed to be going on, instead of reporting the actual state of affairs.

Managers who are used to dealing with computers that have been secured may believe that computers are automatically secure; the shipped configuration will be reasonably safe if it is connected to a network. This is not true. In fact, the truth is almost the exact opposite. The default configuration that machines are shipped with is usually laughably insecure, and it requires considerable expertise to arrive at a secure configuration. Therefore, a manager who says that all of the computers are perfectly secure may be completely incorrect, without having the least intention of deceiving you.

If you ask questions that have clear "right" answers, most people will tend to try to give you those answers. Other people will become defensive. Try to ask neutral questions that don't have a clear bias. For example, don't ask people if they think security is important; instead, ask which is more important to them, security or a cooperative work environment, and then get them to expand on that answer.

When you talk to people, make it extremely clear why you're asking. Asking about security policies tends to give people the impression that you're trying to check up on them. Some people will try to get a good grade, rather than discussing reality. Others will become hostile (after all, why should you be checking up on them?). If you get either of these reactions, stop asking questions about security policies (there's no point in it if they're not going to give useful answers), and go back to trying to explain what you're doing and why. If they never believe you, ask somebody else.

External factors

Your site isn't completely independent. There are issues outside of a computer facility that influence security policy. These include legal requirements, contractual obligations, and existing organizational policies.

Let's look first at legal issues. In the United States, a publicly traded company has a legal responsibility to its shareholders to protect its assets. This means that if you work for such a company, even if everybody at the company agrees that you ought to remove all of the passwords and let the Internet in, you can't choose that as a security policy. Your security policy must show evidence that you are safeguarding the company's computers and information. What's required is "due diligence," an attempt in good faith to take normal precautions. "Normal precautions" limit what you need to do; you don't have a legal responsibility to require retinal scans before people can touch the computers!

Regardless of the type of institution you work for, in most places in the United States there is also a legal responsibility to safeguard certain types of information about employees. Employee reviews are generally legally protected; so are straightforward personnel records of information like home addresses. Universities have legal responsibilities regarding the safeguarding of student records, right down to the information about which students attend the university. Data about individuals has even more legal protection in some European countries. If you do not work for Human Resources or Student Records, you may think you don't have to worry about protecting this kind of information, but you're probably wrong. Usually, every manager or supervisor has confidential employee data to deal with; similarly, the information used to maintain accounts at universities contains confidential student data (e.g., whether or not the student is enrolled, and what classes they're taking).

Your organization may also have contractual obligations to protect data. If you have customer or client data on your systems, your contracts probably require you to protect it. (This may apply to research contracts at universities as well.) If you have source code or prerelease software, you almost certainly have a license that requires you to protect it.

Your organization may also have existing policies that influence security policies. Often, these are policies about the protection of data (usually written to meet the many and varied legal obligations discussed above), but there may be policies requiring that people have access to data, especially at universities and public institutions.

If your organization has a legal department, consult them. (Don't invite them to write up a policy; just ask them to explain the institution's legal obligations.) If your organization does not have a legal department, consult a senior manager. In any case, find any existing written policies and wade through them to see what they say that's relevant to security. Going through these written policies will also give you a good idea for what works and doesn't work in a written policy. If you like the existing policies, base your new ones on them. If you hate the existing policies, resist the temptation to make your new ones like them just because it's the way it's always been done before.

Getting strategic and policy decisions made

Strategic decisions need to be understood and made by top-level management or they will never be successfully implemented. If you don't have top-level management support for security, you aren't going to have security; it's that simple. Why wouldn't you have support from top-level managers? Probably because you haven't addressed their concerns in ways they understand. Here are some things to consider in making your case.

Involve everybody who's affected

You may be the person with the best understanding of the technical issues, but you aren't necessarily the person with the best understanding of the institution's needs as a whole. Strategic and policy decisions must be made by people working together. You can't just come up with a policy you like, take it around to a lot of people, and have them rubber stamp it. Even if you manage to get them to do it—which may well be more difficult than getting them to help make intelligent decisions—they won't actually follow it.

continued on next page

Internet Security Policies *(continued)*

One major computer manufacturer had a policy forbidding dial-in modems. Unfortunately, the company's centralized dial-in access didn't satisfy all of their programmers. Some of these programmers figured out that, although they couldn't request modem lines, they could redirect existing fax lines to modems, go home at night, and dial up their work computers. Even more unfortunately, a programmer in one of the groups with this habit was fired and proceeded to break into the site. He systematically tried all the phone numbers in the range the company had assigned to fax machines until he connected to one of the redirected ones and got a login prompt from an unsecured machine inside the corporate firewall. The former employee did significant damage before he was detected and shut out. He was able to gain a lot of time because the people trying to shut him out didn't know the modems existed. When they did figure out that modems were involved, the process of getting rid of them all proved to be tedious and prolonged, because lines were diverted only when people planned to use them.

That whole incident was the result of the fact that management and system administrators had a policy that ignored some genuine needs of the people using the computer facility. The official policy required dial-in access to be so secure it was almost completely unusable, and the unofficial policy required dial-in access to be so usable that it was almost completely insecure. If there had been a policy that allowed moderately insecure dial-in access, the break-in might have been avoided, and it certainly would have been easier to detect and stop. It would also have been avoided if the programmers had agreed that security was more important than dial-in access, but that kind of agreement is much harder to achieve than a compromise.

In fact, in this case there wasn't much actual disagreement between the parties involved. If the managers had been asked, they would have said that letting people work from home was important to them; they didn't understand that the existing dial-in system was not providing acceptable service. If the programmers had been asked, they would have said that preventing people from maliciously deleting their work was important to them; they didn't understand the risks of what they were doing. But nobody thought about security and usability at the same time, and the result was pure disaster.

Accept "wrong" decisions

You may find that the security policy you come up with is one you don't particularly like. If this happens because the people who made it don't understand what they've done, then you should fight strongly to get it fixed. If, on the other hand, people understand the risks, but they don't share your priorities, put your objections down in writing and go ahead with the policies. Yes, this will sometimes lead to disasters. Nonetheless, if you ask a group to make a decision, you can't insist that it be your decision. You also can't be sure that your way is the only right way.

Sometimes managers have a genuine willingness to accept risks that seem overwhelming to system administrators. For example, one computer manufacturer chose to put one of their large and powerful machines on an unprotected net, and to give out accounts on the machine to customers and prospective customers upon request. The system administrator thought this was a terrible idea and pointed out that the machine was fundamentally impossible to secure; there were a large number of accounts, changing rapidly, with no pattern, and they belonged to people the company couldn't control.

Furthermore, the reason the company was giving out test accounts was that the machine was a fast parallel processor, which also meant that it might as well have been designed as the ultimate password-cracking machine. To the system administrator, it seemed extremely likely that once this machine was broken into (which was probably inevitable), it was going to be used as a tool to break into other machines.

A battle ensued, and eventually, a compromise was reached. The machine was made available, but extra security was employed to protect internal networks from it. (This was a compromise because it interfered with employees' abilities to use the machine, which they needed to do to assist the outsiders who were using it.) Management chose to accept the remaining risk that the machine would be used as a platform to attack other sites, knowing that there was a potential for extremely bad publicity as a result.

What happened? Sure enough, the machine *was* compromised, and was used to attack at least the internal networks. The attacks on the internal networks were extremely annoying, and cost the company money in system administrators' time, but they didn't produce significant damage, and there was little or no bad publicity. Management considered this expense to be acceptable, however, given the sales generated by letting people test-drive the machine. In this case, conflicting security policies were resolved explicitly—by discussion and compromise—and the result was a policy that seemed less strong than the original, but that provided sufficient protection. By openly and intentionally choosing to accept a risk, the company brought it within acceptable limits.

Present risks and benefits in different ways for different people

You need to recognize that different people have different concerns. Mostly, these concerns are predictable from their positions, but some are personal. For example, suppose that:

- Your chief financial officer is concerned about the cost of security, or the cost of not having enough security.
- Your chief executive officer is concerned about the negative publicity a security incident involving your site could bring, or about potential loss or theft of intellectual property via the Internet.
- A department chair is concerned that tenure reviews will be revealed.
- A mid-level manager is concerned his employees are squandering all their time reading USENET news or surfing the Web.
- Another mid-level manager is concerned her employees are importing virus-infected PC software from the Internet.
- Still another mid-level manager is concerned how best to provide technical support to customers over the Internet.
- A professor is concerned her data won't be accessible from other institutions while she's on sabbatical.
- An instructor is concerned that students are stealing answers from each other or tests from instructors.
- Users are concerned about the availability of Internet services they feel are vital for their jobs.
- Users are concerned they won't be able to work together if there are too many security issues.

continued on next page

Internet Security Policies *(continued)*

- Students are concerned they won't be able to play with the computers, which is a part of how they learn.
- Graduate students and project managers are concerned that security measures are going to slow down projects with strict timelines.

You need to take the time to discover all of these different, legitimate concerns and address them. You may also decide there are things that these various people *should* be worried about, but aren't, because they don't know any better; you have to educate them about those issues. This means you need to take the time to understand their jobs, what they want to accomplish with the network, and how well they appreciate the security issues.

Talk to each of these people in terms they care about. This requires a lot of listening before you ever start talking. To managers, talk about things like probable costs and potential losses; to executives, talk about risk versus benefit; and to technical staff, talk about capabilities. Before you go in with a proposal, be prepared with an explanation that suits your audience's point of view and technical level. If you have trouble understanding or communicating with a particular group, you may find it helps to build a relationship with someone who understands that group and can translate for you.

You're not trying to deceive anybody. The basic information is the same, no matter who you're talking to. On the other hand, if a particular decision saves money and makes for a more enjoyable working environment, you don't go to the chief financial officer and say "We want to do it this way because it's more fun," and then go the programmers and say "We want to do it this way because it's cheaper."

Avoid surprises

When it comes to security, nobody likes surprises. That's why you need to make sure that the relevant people understand the relevant issues and are aware of, and agree with (or at least agree to abide by), the decisions made concerning those issues.

In particular, people need to know about the consequences of their decisions, including best, worst, and probable outcomes. Consequences that are obvious to you may not be obvious to other people. For example, people who are not extremely UNIX-knowledgeable may be quite willing to give out root passwords. They don't realize what the implications are, and they may be very upset when they find out.

People who have been surprised often overreact. They may go from completely unconcerned to demanding the impossible. One good break-in, or even a prank, can convert people from not understanding all the fuss about passwords to inquiring about the availability of voiceprint identification and machine gun turrets. (It's preferable to get them to make decisions while they are mildly worried, instead of blindly panicked!)

Condense to important decisions, with implications

When you're asking a top manager to decide issues of policy, present only the decision to be made and the pros, cons, and implications of the various options: not a lot of extraneous decisions. For example, you shouldn't waste your CEO's time by asking him or her to decide whether you should run *Sendmail* or SMAIL as your mailer; that's primarily a technical issue, and one that should be resolved by the relevant technical staff and managers.

On the other hand, you may need to call upon your CEO to decide strategic issues regarding mail, such as whether or not everyone in the organization is to have e-mail access, or only certain people (and if it's to be limited, to whom).

Don't offer people decisions unless they have both the authority and the information with which to make those decisions. Always make it clear why they're being asked to decide (instead of having the decision made somewhere else). In most cases, you want to avoid open-ended questions. It's better to ask "Should we invest money in a single place to be a defense, or should we try to protect all the machines?" than "What do you think we should do about Internet security?" (The open question gives the replier the option of saying "nothing," which is probably not an answer you're going to be happy with.)

Justify everything else in terms of those decisions

All of the technical and implementation decisions you make should follow from the high-level guidance you've obtained from your top managers and executives. If you don't see which way you should go with a technical issue because it depends on nontechnical issues, you may need to request more guidance on that issue. Again, explain clearly the problem; the options; and the pros, cons, and implications of each option.

When you explain policies or procedures, explain them in terms of the original decisions. Show people the reasoning process.

Emphasize that many issues are management and personnel issues, not technical issues

Certain problems, which some people try to characterize or solve as technical problems, are really management or personnel problems. For example, some managers worry that their employees will spend all their time at work reading USENET news or surfing the Web. However, this is not a technical problem, but a personnel problem: the online equivalent of employees spending the day at their desks reading the newspaper or doing crossword puzzles.

Another common example of misdirected concern involves managers worrying that employees will distribute confidential information over the Internet. Again, this usually isn't a technical problem; it's a management problem. The same employee who could e-mail your source code to a competitor could also carry it out the door in his pocket on an 8mm tape (generally far more conveniently and with less chance of being caught). It is irrational to place technological restrictions on information that can be sent out by e-mail unless you also check everybody's bags and pockets as they leave the premises.

Don't assume that anything is obvious

Certain things that seem obvious to a technical person who is interested in security may not be at all obvious to nontechnical managers and executives. As we've mentioned, it's obvious to anyone who understands IP that packet filtering will allow you to restrict access to services by IP addresses, but not by user (unless you can tie specific users to specific IP addresses). Why? Because "user" is not a concept in IP, and there's nothing in the IP packet that reflects what "user" is responsible for that packet. Conversely, certain things that seem obvious to managers and executives are not at all obvious to technical staff, e.g., that the public's perception (which is often incomplete or simply incorrect) of a problem at your company is often more important than the technical "truth" of the matter.

What if you can't get a Security Policy?

What do you do if, despite your best efforts, you can't get a security policy written down? The safest answer is this: document, document, document. Write down what you're doing, and why, and what the existing policies are, and what you tried, and why you think the situation is bad.

continued on next page

Internet Security Policies *(continued)*

Print it out on paper, sign it, and deliver it—at least to your manager, if not to several managers above your manager. File a paper copy, with your signature and the dates you gave it to people.

Every year, or every time there is a significant change in the situation, try to get the policy created again. If it doesn't work, repeat the entire documentation process. Be sure to edit the document; it's tempting to just change the date and resend it, but it probably won't be quite right any more, and it weakens your position.

Doing what we recommend is fairly confrontational behavior, and it can look as if you're more interested in making certain that you're safe than in making certain your site is safe. (This may be true, but it's not going to get anybody to fix anything.)

It's worth working a long time on getting your document to say exactly what you want it to say. Don't fall into the trap of feeling that you have to use formal language. If what you want to say is "I understand that we're an informal company and we don't do written policies, but I think this issue is so important that we still need to have something written down," just say exactly that.

References

[1] Garfinkel, Simson & Spafford, Gene, *Practical UNIX and Internet Security*, O'Reilly & Associates, 1996, ISBN 1-56592-148-8.

[2] Holbrook, P., and Reynolds, J., "Site Security Handbook," RFC 1244, July 1991.

[3] Ranum, Marcus, "Internet Firewalls Frequently Asked Questions (FAQ)," Available from: `http://www.iwi.com/pubs/faq.htm`

[4] Ranum, Marcus, "Thinking About Firewalls," 1993. Available from: `ftp://moink.nmsu.edu/firewalls/fwalls.ps.Z`

[5] See also the `comp.admin.policy` newsgroup.

[6] Doty, T., "The Firewall Heresies," *ConneXions,* Volume 9, No. 6, June 1995.

[7] Doty, T., "A Firewall Overview," *ConneXions,* Volume 9, No. 7, July 1995.

[8] Chapman, D. B., and Zwicky, E. D., "Internet Security Strategies," *ConneXions,* Volume 9, No. 12, December 1995.

D. BRENT CHAPMAN is a consultant in the San Francisco Bay Area, specializing in Internet firewalls. He has designed and built many Internet firewall systems for a wide range of clients, using a variety of techniques and technologies. He is the manager of the Firewalls Internet mailing list. Before founding Great Circle Associates, he was operations manager for a financial services company, a world-renowned corporate research lab, a software engineering company, and a hardware engineering company. He holds a Bachelor of Science degree in Electrical Engineering and Computer Science from the University of California, Berkeley. In his spare time, Brent is a volunteer search and rescue pilot, disaster relief pilot, and mission coordinator for the California Wing of the Civil Air Patrol (the civilian auxiliary of the United States Air Force). E-mail: `Brent@GreatCircle.com`

ELIZABETH D. ZWICKY is a senior system administrator at Silicon Graphics, and the president of the System Administrators Guild (SAGE). She has been doing large-scale UNIX system administration for 10 years, and was a founding board member of both SAGE and BayLISA (the San Francisco Bay Area system administrators' group), as well as a non-voting member of the first board of the Australian system administration group, SAGE-AU. She has been involuntarily involved in Internet security since before the Internet Worm. In her lighter moments, she is one of the few people who makes significant use of the "rand" function in *PostScript,* producing *PostScript* documents that are different every time they're printed. E-mail: `zwicky@sgi.com`

[Ed.: This article is adapted from *Building Internet Firewalls* by D. Brent Chapman and Elizabeth D. Zwicky, published by O'Reilly & Associates, 1995, ISBN 1-56592-124-0, 1-800-998-9938. Used with permission.]

Back to Basics:

Networking requirements of audio and motion video

by François Flückiger, CERN

Introduction

With the exploding development of distributed multimedia applications, existing networks—such as the Internet or conventional shared-medium LANs—are increasingly requested to carry new types of traffic. Thus, active work is underway to improve and enhance these networks (RSVP and IPv6 in the Internet, Switched, Fast, and Isochronous Ethernet, FDDI-2...). In parallel, ATM is proposed as *the* solution to integrate all types of services. Audio and motion-video are two media which place some of the most stringent requirements on the underlying network. In this article, we analyze some of these requirements, both in quantitative and qualitative terms.

In multimedia terminology, sound and moving images are called *continuous* media, as the presentation of the information requires a continuous playout as time passes. In other words, time, or more exactly time-dependencies between information items, is part of the information itself.

Downloading versus real-time transmission

Audio and/or motion video are used in three broad types of networked applications: interpersonal communications such as telephony, videoconferencing or multimedia desktop collaboration, distribution applications such as the Internet broadcast over the Mbone network and server-based applications such as the remote access to audio or video servers.

For each of these applications, the transfer of audio or video sequences may be performed either by downloading or in real-time mode. In the former mode, the information, or part of it, is first transferred, then stored at the receiving end, and further displayed. In the latter, part or all of the information is transferred in real time over the network for on-the-fly presentation on the receiving system.

In practice, downloading is generally reserved for small sequences that can easily be stored, even in the central memory of the receiving system. The playback may take place automatically straight after completion of the transfer. With audio, the user accessing a server may even have the impression that the server is reacting in real time to the request, if the bit rate of the network is sufficient and the sound sequences are actually small. Note that this real-time feeling is difficult to quantify, as it depends on the subjective perception the user has of the complexity involved in his or her request—user tolerance to delays is higher when the user perceives the request as requiring a highly complex computational or data communication process. Interpersonal audio—such as in packet telephony—may also employ some kind of downloading, where an entire phrase is first captured and then sent as a data block to be played out after complete reception. In fact, the requirements that audio/video downloading places on the network are of the same nature as those of other asynchronous data set transfers.

In the case where the sound or motion video sequence is too long, or the network too slow, real-time transmission is necessary. This creates much more stringent demands on both the underlying network and the sink system.

continued on next page

Network requirements of audio and video *(continued)*

The above model of two transmission modes is a simplification of the reality. In practice, the lines have blurred and the solutions employed lie on a continuum. Indeed, there is no strict "real-time playout" per se, as any sequence of sound or video needs to be buffered in the memory of the sink system, for "delay equalization" or decompression purposes.

Audio should have higher priority

The psycho-acoustic behavior of our ear may be modeled as a *"differentiator."* We know that the hearing process has considerable powers of discrimination. For example, the mechanisms by which we can recognize the semantics of a given conversation when mixed with two others are extremely complex and produce remarkable results. In contrast, the mechanism of vision acts as an *"integrator."* For example, it may be extremely difficult to recognize two or three intermixed drawings and interpret them.

The consequence of the above discussion is simple: humans are much more sensitive to alterations of audio than of visual signals. As a result, our tolerance of transmission errors affecting audio streams is much lower than our tolerance of errors affecting motion video streams. When an audio and a video stream are to be transmitted concurrently, some networks may allocate clearly separate channels to each stream, others such as most packet networks mix the two. As a result, in many packet networks, the two streams will compete for the same resources. In such cases, the audio streams must have priority over the video stream, as far as this makes sense for the network.

Priority mechanisms exist, with varying degrees of sophistication in Frame Relay (discard eligibility mechanism), the future IPv6 (IP flows with service quality), ATM (in fact, via quality of service associated with virtual connections), ST-II (again over virtual links), synchronous FDDI and the 100VG-AnyLAN version of 100Mbps Ethernet. Other technologies do not need a concept of priority per se as they implement a split of the bandwidth into separate channels. They include ISDN, IsoEthernet (the combination of conventional 10Mbps Ethernet with 96 ISDN circuits over a single 10Base-T connector), and to some extent FDDI-2.

Audio requirements

Let us turn to more quantitative requirements. There is no single answer to the question: "Which bit rate is required to carry sound?" because there is no single sound quality. The quality of the sounds produced by computer-mediated systems lies on a continuum, ranging from that of low-end speakers in personal computers to three-dimensional studio-quality sounds. On this continuum, let us consider two typical levels, speech telephony quality and compact disk audio quality.

Speech telephony quality is the one that you experience when using your regular telephone. There are various ways of digitally encoding and possibly compressing speech analog signals. The most conventional technique has been standardized by the ITU for the support of digital telephony over voice-grade cables and is called *G.711*. This standard has a bandwidth of 3.4kHz. This means that frequencies outside this band will be either filtered or seriously attenuated. When speaking normally, humans produce sounds with a spectral composition which roughly occupies a 10kHz band.

PCM: simple, but not optimal

The G.711 standard is based on a *Pulse Code Modulation* (PCM) digitizing algorithm using *logarithmic* coding. PCM is the simplest technique to digitize an analog signal.

One important characteristic is that PCM is not specific to a particular type of signal. In particular, PCM is not speech specific. Any analog-to-digital conversion comprises two steps: the sampling—or time discretization—step, and the quantization—or amplitude discretization—and code-word generation step. The characteristic of PCM is that "the analog signal is sampled, and each sample is quantized independently of other samples and converted by encoding to a digital signal" (ITU G.701).

In PCM, there are various ways of quantizing the value of each sample and then associating a code-word to each quantum. Two of these techniques widely used in audio coding are *linear* coding and *logarithmic* coding. Linear coding consists of taking the scaled value of the sample as the code-word. Thus, this is a straightforward technique, though it is not the most widespread one used in telephony. With logarithmic coding, the difference is that the scaled value of every sample undergoes a logarithmic transformation and the result is then represented with binary digits. Thus, the code-word contains the sign and the logarithms of the scaled value of the sample. This may sound unnecessarily complicated, but is due to certain properties of the perceived quality of speech signals where high-amplitude values can be represented with a lower accuracy than lower-amplitude values. Another advantage is that the signal-to-noise ratio is more uniform. There are two types of logarithmic transformations defined by the G.711 ITU standard. The so-called *μ-law* transformation is used in North America and Japan. The *A-law* transformation is used in Europe and in other parts of the world.

Nyquist

The Nyquist theorem indicates that to faithfully represent an analog signal of maximum frequency *f*, the sampling rate should be equal or superior to *2f*.

As the targeted bandwidth—that of voice-grade wires—was in the order of 3.5kHz, the sampling rate was chosen to be equal to 8kHz. The amplitude of each digitized sample in PCM G.711 digital telephony is represented with 8-bit code-words. The benefit of logarithmic coding may be illustrated by the fact that with linear coding, 14 bits instead of 8 would be necessary to obtain an equivalent perceived quality. The bit rate of the standard digital telephony we know is therefore 64Kbps. This is compatible with most reasonably loaded shared-medium LANs such as Ethernets. However, not all pairs of end-systems connected to the wide-area Internet may benefit from a sustained 64Kbps bit rate.

Other ITU speech standards

The ITU has defined many other standards based on digitizing techniques differing from the simple PCM logarithmic techniques. The G.721 standard generates a 32Kbps stream without noticeable degradation of the perceived quality. G.721 is based on the *Adaptive Differential Pulse Code Modulation* (ADPCM) technique. There, only the difference between samples is coded—more exactly, this the difference between the current value and a predicted value which is calculated from previous samples by means of a function which varies with time. As in G.711, the sampling rate is 8kHz, but each value is coded with 4 bits only. G.722 is a more sophisticated standard which targets improved quality—7kHz bandwidth instead of 3.4 with the other schemes—at 48, 56 or 64Kbps. In the standard mode, the sampling rate of G.722 is 16kHz and the amplitude depth is 14 bits. It is based on a sub-band ADPCM method. G.728 is another standard targeting low bit rate.

continued on next page

Network requirements of audio and video *(continued)*

It operates at 16Kbps only, for a bandwidth limited to 3.4kHz, but the resulting quality is inferior to that of the other standards. However, G.728 is widely used in particular for audio/videoconferencing. It is well suited over basic rate ISDN connection as well as over the Internet, either in broadcast or interactive applications. However, residential users connected via 14.4Kbps modems cannot use these schemes.

Other speech compression techniques

Other compression schemes exist for speech. GSM is a compression standard for mobile telephony used in Europe in particular. Version 6.1 operates at 13.2Kbps. The sampling rate is 8kHz. The resulting quality is inferior to that of G.711 of G.722 systems.

But the most highly performing schemes use sub-band coding and *vector quantization*. Vector quantization is a generic technique which consists of dividing the bit stream to be compressed into blocks called *vectors* and replacing each block with the number of an entry in a table called a *code-book*. The code-book contains patterns and the transmitted number corresponds to the entry of the best matching pattern in the table. The arithmetic difference between the actual data block and its best matching pattern may also be transmitted. The code excited *linear prediction* (CEL-P) technique, operating at 4.8 Kbps for a quality slightly inferior to conventional G.711 telephony, is used in the US Federal Standard 1016. Another standard, Federal Standard 1015, uses *linear predictive coding* (LPC) and operates at 2.4Kbps. The latter results in an intelligible but somewhat artificial voice. CEL-P and LPC are particularly well suited for speech over the Internet when the end-to-end bit rate is very low—because of internal congestion or when systems are connected via modems.

CD-quality bit rate

Speech is a sound, but not all sounds are like speech. The range of frequencies we can generate is narrower than that we can detect. Also, in speech, repetitive patterns may more easily be found. And during a conversation, hopefully not all participants speak at once. These properties have been exploited by coding and compression systems. Thus speech may be up to 10 times less demanding than, say, music in terms of bit rates.

The home audio compact disks (*CD-audio*) aim at covering the audible frequency range of humans, that is about 20kHz. Thus, according to the Nyquist theory, the sampling rate of the analog sound should be at least 40kHz. In practice, the sampling rate is 44.1kHz. The coding uses a PCM scheme, but unlike voice-grade digital telephony, in CD-audio the coding of the samples follows a *linear* algorithm. The amplitude of each sample is represented with 16-bit code-words instead of 8-bit in conventional PCM telephony.

Thus, uncompressed CD-quality audio requires 705.6Kbps for one monophonic channel. As compact disks are stereophonic, the bit rate required from a network to transmit a full stereophonic sound in CD quality is 1411.2Kbps. This bit rate is compatible with the throughput of the popular T1 (DS-1, 1.544Mbps) digital leased circuits. But it is also possible to listen to CD-quality classical music over unloaded Ethernet segments or high bandwidth portions of the Internet. Note that CD quality is different from *sound studio quality*. Studio quality usually requires twice the bit rate of CD quality.

Though digital signals on CD-audio are not compressed, several compression techniques exist for CD-quality sound. They are used for hi-fi audio transmission or in audio-video broadcasts.

One of them, the *MUSICAM* scheme, has been adopted for one of the layers of the MPEG standard (Layer-2). A stereophonic CD-quality sound compressed with MUSICAM requires in the order of 192 to 256 Kbps. MPEG-Audio Layer 3—a more recent option—is the most performing scheme of the series and targets near-CD-quality sound at 64Kbps only per monophonic channel.

Transit delay for audio streams

Bit rate is one type of requirement. Network latencies is another. The requirements on the transit delay for the real-time transmission of audio streams are highly dependent on the multimedia applications. In pure distribution (unidirectional transmission), this delay may be almost as long as the technology allows it to be, up to seconds. This is not the case of course with interactive applications, such as a conversation between people, the consultation of remote information upon voice input, or response to voice in virtual reality.

For conversation between people, a technical difficulty lies in the echo that may be audible if (a) the end-to-end return trip delay exceeds a certain value, and (b) no particular measure is taken to limit the echo such as the use of directional microphones and speakers or the use of echo canceling systems. The ITU-T has defined 24ms as the upper limit of the one-way transit delay beyond which echo canceling techniques have to be employed.

For applications where a response is expected from a system after voice input, the round-trip delay should generally stand between 200 and 1000 ms, which requires one-way transit delays below 100ms to 500ms. In virtual reality where an impression of immersion is requested, the feedback should happen less than 100ms after the input, which gives a required network transit delay in the order of 40ms.

Jitter: the main source of problems

The delay *jitter* is the variation over time of the network transit delay. This is an essential performance parameter of the network intended to support real-time sounds. Basically, if blocks of information carrying sounds arrive after widely varying transit delays, the only solution to overcome this is for the receiving system to wait a sufficient time, called the *delay offset,* before the playout, so that most of the delayed blocks are given a chance to arrive in time. Incoming blocks are stored in a buffer. This process is sometimes called *delay equalization.* This technique may add a substantial component to the overall latency between the source and the final playout of the sound.

Again, the jitter toleration is mainly dictated by the maximum overall delay that the particular application can accept. In interactive applications, the maximum tolerated jitter is usually set at 100 to 800 ms.

Note that if blocks of sound—that is in practice, packets—are missing, either because they never arrived or they arrived after the delay equalization, all is not lost. The missing information may be extrapolated or interpolated from received blocks. This is called *error concealment.* In certain cases, the end-user may hardly notice that a data loss has such been recovered.

Motion video: several classes of quality

We now turn to the requirements needed for the transport of real-time video. As for sounds, this depends on the video quality. We may consider five classes of quality, which in fact provide five distinct points on a continuum: High-Definition TV (HDTV), studio-quality digital TV, current broadcast-quality TV, VCR-quality, and low-speed videoconferencing quality.

The quality of motion video is essentially perceptual and can only be measured by experimental techniques such as *mean opinion scores* (MOS).

continued on next page

Network requirements of audio and video *(continued)*

However, four technical parameters may affect the overall perceived quality of motion video. They are:

- The spatial resolution—the number of pixels per frame,
- The color or chroma resolution—the number of bits to code each pixel,
- The temporal resolution—the number of frames per second,
- The scanning mode—either interlaced or progressive.

Several resolution levels, frame rates, and two scanning modes have been proposed and standardized for HDTV. The highest quality implies high resolution (that is 1920 pixels per line and 1080 lines per frame) and high frame rate (60 frames per second). Sending uncompressed HDTV over a network would require a bit rate in the order of 2Gbps. This exceeds the capacity of existing production networks. The MPEG-2 standard has defined a compression scheme (the "high level") which, when implemented, could deliver HDTV at 25 to 30 Mbps. Few networking technologies can deliver such bandwidth easily. They include high-speed ATM virtual connections, FDDI or Fast (100 Base-T) Ethernet with very few stations per segment, and Switched 100Mbps Ethernet. But with any of these technologies, only a few number of concurrent HDTV streams can be supported.

Motion video: from 1.2 to 7 Mbps

Studio-quality digital television has been defined by the ITU-R in recommendation 601 in the middle of the 1980s. The idea was to standardize practices and techniques in use as early as 1972 in televisions studios, essentially for digital video effects. Studio-quality digital TV is produced either by digitization of the analog signals from a conventional video camera, or directly by a digital camera. The frame format is 720 pixels per active line and 525 or 625 lines per frame depending on the NTSC or PAL/SECAM option. Each pixel is coded with 24 bits. The ITU-R 601 standard defines the frame rate as equal to that of the current broadcast analog standard: that is 30 fps for NTSC and 25 fps for PAL/SECAM. As a result, an uncompressed studio quality video stream requires 166Mbps. This is too much for most networking technologies. A simple compression technique consists of compressing individually each frame independently of the other. The most popular scheme relies on the JPEG standard, a technique to compress continuous-tone still images. On the contrary, the MPEG standard uses in addition a mechanism of motion compensation where redundancies between images are searched and reduced. The version 2, MPEG-2, allows compression of studio-quality video down to 4 to 7 Mbps, depending on the implementation.

Broadcast quality is inferior to studio quality; the NTSC spatial resolution especially is significantly lower. Note also that conventional broadcast TV, as well as the ITU-R 601 digital standard use an interlaced scan where each frame is divided into two fields each dealing with one every two lines. Computer displays usually use progressive scan. At equal bit rate progressive scan gives a significantly better perceived quality. Early implementations of MPEG-2 provides PAL/SECAM broadcast quality—that is, a resolution of about 540 × 480 samples/frame—at a rate of 6Mbps. At resolution which is close to that of NTSC, the achieved bit rate is in the order of 3Mbps and is expected to be reduced to 2Mbps. Such compression rates allow the transmission of broadcast-quality TV over regular copper telephone lines using the *Asynchronous Digital Subscriber Loop* (ADSL) technology.

VCRs have a lower resolution than broadcast TV. The MPEG-1 compression standard was targeting VCR quality at 1.2 to 1.4 Mbps, that is a bit rate compatible with that of T-1 leased lines.

Finally videoconferencing quality refers to a *Common Intermediate Format* (CIF) image (352 × 288) and a frame rate in the order of 5 to 10 fps. Videoconferencing typically requires 112Kbps with today's ITU-T H.261 compression scheme or the equivalent. But products are emerging for mobile videophony using GSM technology at 32Kbps.

The 2000 Olympic Games

Over the ever-increasingly complex networking infrastructure—often a mixture of various LANs and long-distance Internet networks—the major difficulty in the support of continuous media lies in guaranteeing end-to-end bit rates and bounded transit delays. Unpredictable packet losses and excessive jitter are the problem. As a result, research is active in two fields: provide level of service guarantees—that is, acting at the level of the network—and recover from the network deficiencies within the end-systems (adaptive buffering, extra- and interpolation, adaptive resolution, ...). These two approaches, sometimes viewed as two distinct schools of thought, are in practice complementary and their combination may contribute to the realization of Nick Lippis's prediction (in his keynote speech at NetWorld+ Interop Conference, Las Vegas, 1995): "100 million people will view the 2000 Olympic Games over the Internet."

Small dictionary

ADPCM *Adaptive Differential Pulse Code Modulation.* A predictive-encoding technique derived from PCM, which only encodes the difference between the actual value of the sample and a predicted value which results from some form of extrapolation of the preceding values using a variable prediction function.

CIF *Common Intermediate Format.* Interchange format for motion video images. The frame size for the luminance is 352 samples per line and 288 lines per frame. Each color difference component contains 176 samples per line and 144 lines per frame. The sampling ratio is noted 4:1:1. One of the formats supported by the ITU-T H.261 recommendation for videoconferencing.

Continuous media Term used to refer to audio and motion video media. In continuous media, time is part of the semantics. Continuous media are also called time-dependent media or time-based media. Opposite of *discrete media.*

Digitization Process by which an analog signal is transformed into a digital signal. Digitization involves two steps: the sampling of the analog signal, that is its transformation into a discrete suite of values taken at a given time or space, and the quantization and code-word generation.

G.711 Standard from ITU-T for the encoding of digital speech designed for standard digital telephony. Based on a PCM digitizing algorithm using logarithmic encoding. Targets bandwidth in the order of 3.5kHz. Sampling rate respects Nyquist theory and is equal to 8kHz. The amplitude of each digitized sample is represented by 8-bit code-words. The bit rate of G.711 digital signals is therefore 64Kbps.

G.721 Standard from ITU-T for the encoding of digital speech. Generates a 32Kbps stream. Based on the ADPCM technique. Each value difference is coded with 4 bits. As in G.711, the sampling rate is 8kHz.

G.722 Standard from ITU-T for the encoding of digital speech. The objective is to provide better sound quality than the conventional G.711 PCM scheme or the G.721 compression technique.

continued on next page

Network requirements of audio and video *(continued)*

Based on a sub-band ADPCM scheme (SB-ADPCM). The bandwidth of a G.722 compressed signal ranges from 50Hz to 7kHz. The resulting bit rate is 48, 56, or 64Kbps—in the standard mode, the sampling rate is 16kHz with 14-bit amplitude depth.

G.728 Standard from ITU-T for the encoding of digital speech. Targets low bit rates. It operates at 16Kbps with a bandwidth limited to 3.4kHz. The resulting sound quality is significantly inferior to that of G.711 or G.722. Based on a scheme called *low-delay code excited linear prediction* (LD-CELP).

H.261 Standard from ITU-T for videoconferencing at low and medium bit rates. H.261 addresses motion video encoding and compression aspects. It relies on the discrete cosine transform, DPCM, and motion compensation. Defines two frame formats, CIF and QCIF. Designed to operate optimally at a few hundred Kbps, but covers the range 56 to 1920 Kbps. Part of the ITU-T H.320 family of standards for teleconferencing.

Interlacing The technique used in conventional analog television consisting of scanning individual frames into two successive fields, each formed of respectively the odd and even lines of the frame. Interlacing may be involved at sampling, transmission, and playout time. Opposite of *progressive scanning.*

ITU-R 601 Standard from ITU-R—formally called CCIR—for studio-quality digital television targeting broadcasting services. The recommendation defines a frame structure of 525 or 625 lines, to be compatible with either the American NTSC analog television standard, or the European PAL and SECAM standards. Defines a single number of samples per active line: 720. The sampling frequency is 13.5MHz for the luminance signal and 6.75MHz for each color difference signal. Each sample is linearly encoded with eight binary digits. If not compressed, an ITU-R 601 digital signal generates a bit rate of 165Mbps.

Jitter In data communications, the variation over time of the network transit delay. Name initially given to the component of the variation of the transit delay—called the *physical jitter*—which is only due to the transmission equipment. Its meaning has been extended to cover the overall variation, regardless of its origin

JPEG *Joint Photographic Expert Group.* Name of a standard from ISO for continuous-tone still image coding and compression. JPEG offers four modes of operation: sequential coding where the image is encoded in a single left-to right, top-to-bottom scan; progressive coding where multiple scans are involved; lossless encoding where the original signal is restored after decompression; hierarchical encoding, where the same image is encoded several times with differing resolution. In lossy mode, the compression ratio can reach "30 to 1" with no noticeable degradation of quality. JPEG is also used to compress, frame by frame, motion video. This is referred to as *motion JPEG.*

MPEG *Motion Picture Expert Group.* A family of ISO standards for coding and compression of digital motion video and associated audio. Includes MPEG-System, MPEG-Video, and MPEG-Audio. Three distinct purposes and quality levels are addressed by three different versions: MPEG-1, MPEG-2, and MPEG-4.

MUSICAM Audio encoding and compression standard developed by CCITT, IRT, and Philips, and adopted by the MPEG-1 standard for audio and moving picture.

MUSICAM makes use of spectral, sub-band coding, temporal masking, and fast Fourier transform techniques. A monophonic signal may be compressed at 96Kbps for CD-audio quality and 192Kbps for studio-quality sound.

NTSC *National Television Standards Committee.* Standard for video color encoding ("TV standard") used in North America and Japan.

Nyquist sampling Sampling technique which respects the Nyquist theory. The Nyquist classical theory requires that, to faithfully represent an analog signal, the sampling frequency should be equal to or greater than twice the highest frequency contained in the sampled signal. Studies have, however, shown that under certain circumstances, lower sampling frequencies can in practice be used. This is called *sub-Nyquist sampling.*

PAL *Phase Alternate Line.* The standard for video color encoding used in most parts of Europe. See `http://www.bbc.co.uk/aberdeen/tech.htm`

PCM *Pulse Code Modulation.* Simple digitizing technique which consists of a sampling step followed by a quantization and code-word generation step, where each sample is quantized independently of other samples. Particularly used for digitizing speech. The ITU G.711 encoding scheme employs the PCM technique.

Pixel depth In digitized images, the number of bits used to code the attributes of individual pixels. In color images—where each pixel has three attributes—8 bits or 24 bits are examples of pixel depth. The pixel depth defines the amplitude resolution of the image.

Progressive scanning The technique used in computer displays consisting of handling a single field per frame. Each frame is displayed progressively line by line, each line being scanned from left to right. Opposite of *interlacing.*

Real-time transmission In document access or transfer, refers to the technique where a remote document, or part of it, is transmitted as a stream and presented on-the-fly as the reception goes on. Opposite of *downloading.*

SECAM *Systeme En Coleur Avec Memoire.* The standard for video color encoding used in most of Eastern Europe as well as France, the former Soviet Union, parts of Africa and areas in the Middle East.

Transit delay In data communications, the time elapsing between the emission of the first bit of a data block by the transmitting end-system and its reception by the receiving end-system.

Vector quantization A source-encoding compression technique. The data stream is divided into blocks called vectors. Each vector is compared to vector patterns contained in a table called the code-book, to select the best matching pattern. Once found, its reference in the code-book is transferred. Depending on the implementation, the difference between the actual vector and the pattern may be calculated and sent. Used for image and sound compression.

[Ed.: This article is adapted from *Understanding Networked Multimedia,* by François Flückiger, Prentice Hall, 1995, ISBN 0-13-190992-4. Used with permission. See also book review in *ConneXions,* Volume 9, No. 10, October 1995.]

FRANÇOIS FLÜCKIGER is deputy leader of the Communications Systems group at CERN—the European Particle Physics Laboratory, and birthplace of the World-Wide Web. He is also a part time professor at the University of Geneva. Before joining CERN in 1978, he was employed for five years by SESA, Paris, where he contributed to the design and the implementation of the Transpac network. He is one of the pioneers of Internet in Europe, a regular lecturer, and a frequent speaker at academic and professional conferences. He has 22 years of experience in design, development and management of large-scale data networks. He graduated from the École Superieure d'Electricité in 1973 and the Institut d'Administration des Entreprises, Paris. E-mail: `fluckiger@vxcern.cern.ch`

Book Reviews

The World-Wide Web: Beneath the Surf, by Mark Handley and Jon Crowcroft, UCL Press, 1995, ISBN 1-85728-435-6, paperback.

My immediate reaction to *any* book on the World-Wide Web (W3) has become "(sigh) do we really need another?" And since the authors have more than a modest reputation within the Internet community, I even questioned why Mark and Jon would compromise their integrity or waste their collective talents on producing "yet another Web book."

Web books

The answer lies in the title. You have to go beneath the surf (in the U.S., we might say "under the hood," or in England, "under the bonnet") to appreciate *Beneath the Surf.* Web books fall into two categories. The first category is the HTML-primer-how-to-publish-on-the-NET, among which I find Ed Tittel's *HTML for Dummies* amply practical. The second is generally more offensive, characterized by long lists of "way cool Web sites" which have probably long since ceased to be cool by the time the book hits the shelves. Handley and Crowcroft depict books of the second form as guides for cyber-tourists, and observe that "virtual fashions and weather change too rapidly for a book to be a valid medium to store [that] kind of information."

Organization

Beneath the Surf explains how the Web works in a crisp, concise manner. [1] The first two chapters blitzkrieg you through Internet and information server technology, explaining the concepts and terminology of the Internet and in so doing, laying the stage for an introduction to W3. Chapter 3 takes you through a top-down examination of how the Web works and introduces all the important aspects of W3 technology—uniform resource locators (URLs), *HyperText Markup Language and Transfer Protocol* (HTML and HTTP, respectively)—and through simple examples, demonstrates how a Web page is borne. Chapter 4 examines client (browser/navigator) aspects of the Web in detail, while Chapter 5 attends to the details of how to present information using Web servers. Chapters 6 and 7 examine Web page design and linkage issues by illustrating examples of academic and commercial pages rather than by listing URLs. Chapter 8 lists all of the WWW servers known at the time of publication, and describes some configuration aspects of the popular WWW servers. (The approach here is value-neutral; if you are interested in opinions about the strengths and weaknesses of PC, UNIX, and Macintosh servers, I'd again recommend Ed Tittel's *HTML for Dummies.*)

Complete and compact

Chapters 9 and 10 discuss problems with the WWW, and speculate on the future of the Web and the Internet at large. The authors clearly separate themselves from the rest of the pack by identifying the many areas for improvement, and by expressing without hysteria many of the issues that should be of concern to anyone deeply committed to the future of the Internet. A very complete set of appendices present HTML/SGML, URL, HTTP, and MIME grammar and specification.

The book reminds me of a very good compression program: Handley and Crowcroft cram a whole lot of information into a very small package.

—David M. Piscitello, Core Competence, Inc.
dave@corecom.com

References

[1] Mark Handley & Jon Crowcroft, "The World-Wide Web: How Servers Work," *ConneXions,* Volume 9, No. 2, February 1995.

[2] Peter Neumann, "Is Our Network Infrastructure Sound Enough?" *ConneXions,* Volume 9, No. 11, November 1995.

(Above articles are adapted from the books being reviewed here).

Computer Related Risks, by Peter G. Neumann, Addison-Wesley, 1995, ISBN 0-201-55805-X, 367 pages, paperback.

Who are you?

There are two possibilities. First you are an aficionado of the RISKS mailing list—you don't have to read any further. You know what I am going to say. The second possibility is that you are new to the world of RISKS and want to find out more. Why are you reading this then? Go now and buy this book. When you have read it you will realise that when counting possibilities, two is only an approximation to some other, indeterminate and probably much larger number. But don't worry. You are reading this on paper so nothing can go wrong. Maybe.

You should have by now realised that this is going to be what is usually referred to as a "rave review." Let's face it I'm biased—I am a long term RISKS addict and I prepare the World-Wide Web version of the list. You can always trust the verdict of a pusher who is also a user. Well, about as much as you can trust most software, as you will find when you read this book.

Leading expert

Peter Neumann in addition to being a true Renaissance man and a wonderful human being is also the world's leading expert on computer risks. He has been moderating the RISKS mailing list for many years, it having reached Volume 17 at the present moment, and in this book has distilled and analysed the enormous body of experience that this represents. The contributors to the mailing list include some of the most respected figures in Computer Science and also plain, ordinary people with an illuminating story to report. Everything that makes it to the list passes through Peter as well as all the stuff that gets spiked. The end result is that he tends to be better informed than almost anyone else about software or hardware problems that might impinge on people's safety and well-being.

Examples

Much of the book is taken up with pertinent examples, all carefully classified and organised so that related stories are together. Everything is here from well known examples like the Internet Worm and problems with the Intel 486 (the Pentium errors arrived too late to make it into this edition), through airplane crashes to elevator door accidents. Misconceptions are cleared up—the famous FORTRAN DO loop error occurred on Project Mercury not Mariner I, and was found before it could cause any trouble—and there is the odd nod to apocrypha because they always contain some germ of truth.

Humour

All this information is tied together with just enough of Peter's punning humour and apposite quotations to make this book excellent both for light reading and as a source of serious technical material. If I were teaching a course on computer related risks I would recommend no other textbook but this. Each chapter finishes with a set of Challenges which could be used in many contexts and which help to bring out the essential points that have just been covered.

Read it

The final chapters of the book pull everything together and talk about what we need to do to prevent, or at least minimise, the risks inherent in computerised systems. If you are at all involved in any aspect of software or hardware development these chapters are essential reading, though only of course after you have read all the chapters leading up to them. If you are a grunt programmer, make sure your manager has read this book. Managers put this in all your programming teams libraries. Read it. Whatever your background you will learn an immense amount from this book. Oh yes, and I forgot to say. I checked the index. I'm in there.

—Dr. Lindsay F. Marshall, University of Newcastle upon Tyne
lindsay.marshall@newcastle.ac.uk

Call for Papers

Topics

The *18th Biennial Symposium on Communications* will be held in Kingston, Ontario, Canada June 3–5, 1996. This symposium is intended to provide a forum for engineers and researchers in the area of Communications and Signal Processing. Papers are encouraged from new areas of research on communications, as well as those traditionally associated with this conference. Topics of interest include:

- Speech/Image Processing
- Signal Processing
- Satellite Communications
- LAN Interconnections
- Spread Spectrum Techniques
- VLSI in Communications
- Error Control Coding
- Switching Techniques
- Mobile Communications
- ATM Networks
- Computer Communications
- Personal Communications
- Optical Communications
- Multimedia Communications
- Neural Networks
- Communications Software
- Cryptography and Security
- Cellular Systems

Submissions

Papers in other related areas are encouraged. Plenary lectures are planned and accepted papers will be published in the Conference Proceedings (only preregistered delegates will have their papers published in the Proceedings). The sessions will be held on the campus of Queen's University, Kingston, Ontario, Canada. Five copies of an Abstract and a thousand-word (1,000) Summary should be submitted by January 22, 1996 to:

Dr. Hussein T. Mouftah, Symposium Chair
Department of Electrical and Computer Engineering
Queen's University
Kingston, Ontario K7L 3N6
CANADA
Phone: +1 613-545-2934
Fax: +1 613-545-6615
E-mail: `mouftah@eleceng.ee.queensu.ca`

For more information

Symposium on Communications
PO Box 1570, 190 Railway Street
Kingston, Ontario K7L 5C8
CANADA
Tel: +1 613-531-9210
Fax: +1 613-531-0626
E-mail: `events@adan.kingston.net`

Call for Papers

The *International Journal of Computer Systems Science & Engineering* (IJCSSE) will publish a Special Issue on Asynchronous Transfer Mode (ATM) Switching in the third quarter of 1996. The Guest Editors are Hussein T. Mouftah, Queen's University, Canada and Mohammed Atiquzzaman, Monash University, Australia.

Topics

During the past decade, a considerable amount of effort has been made in studying and designing ATM switches which is believed to be the most developed aspect of ATM. The field has now become a mature area and ATM switches are becoming commercially available. This special issue will include a set of original and survey articles from both industry and academia that represents the current state-of-the-art in ATM switching. Topics include, but are not limited to:

- Switch architectures
- Fault tolerance
- Buffering schemes
- Congestion control and traffic management
- Performance modeling
- Practical experience and field trials
- Buffer management
- Simulation techniques for large switches
- Commercial switches

Submissions

Five copies of complete manuscripts (not to exceed 25 double-spaced pages) should be sent to Mohammed Atiquzzaman. Please include a title page containing author(s) names and affiliations, postal addresses, e-mail addresses, telephone numbers, and fax numbers. Electronic (*PostScript* only) submissions are encouraged. Authors should follow the IJCSSE manuscript submission format available from URL: **http://www.eng.monash.edu.au/~atiq**

Mohammed Atiquzzaman
Department of Electrical and Computer Systems Engineering
Monash University
Clayton 3168
Melbourne
AUSTRALIA
Tel: +61 3 9905 5383
Fax: +61 3 9905 3454
E-mail: **atiq@eng.monash.edu.au**

Dr. Hussein T. Mouftah
Department of Electrical and Computer Engineering
Queen's University
Kingston Ontario K7L 3N6
CANADA
Tel: +1 613-545-2934
Fax: +1 613-545-6615
E-mail: **mouftah@eleceng.ee.queensu.ca**

Important dates

Deadline for receipt of manuscripts: February 1, 1996

Notification of acceptance/rejection: April 30, 1996

For more information

IJCSSE is published by CRL Publishing, London, UK. Please contact the editor-in-chief Prof. Tharam Dillon (**tharam@latcs1.lat.oz.au**) for queries regarding the journal, and contact J. Thompson (**100113.2636@compuserve.com**) for sample copies.

Preliminary Announcement and Call for Papers

Background

TINA '96: "The Convergence of Telecommunications and Distributed Computing Technologies" will be held in Heidelberg, Germany September 3–5, 1996. The long-predicted convergence of telecommunications and distributed computing technologies is occurring. Researchers in distributed computing are experimenting with digital switching hardware and developing the operating systems and middleware infrastructure needed to exploit these new capabilities. Developers of advanced telecommunications services are challenged by the need to expand beyond traditional services to embrace new information services and paradigms that combine mobility, extreme performance requirements, multimedia, guarantees of security and reliability, and customizable "virtual" network mechanisms. Meanwhile, public enthusiasm for Internet services is imposing de facto standardization on these emerging systems, while also creating a potentially immense market for a new generation of commercial communication services. It is time for researchers in distributed computing to join forces with those who are developing the emerging telecommunications infrastructure on which new information services will operate. TINA, the preeminent research forum of the telecommunications industry, is therefore expanding its scope to embrace all aspects of distributed computing, including object-oriented systems, multimedia, distributed data management and large-scale database systems, groupware, telecommerce, distributed systems fault-tolerance and network security.

Topics

The conference solicits a mixture of presentations based on real systems that are in ongoing use, basic research on communications software architectures, protocols, and abstractions in support of advanced distributed middleware environments. Although TINA '96 will stress experimental results and experience with real systems, theoretical contributions of clear practical importance will also be considered. Topics addressed from previous TINA conferences include:

– *Telecommunications Network Architectures:*
- Architectural models for distributed computing & communication
- Architectures for network management and intelligent networks
- Distributed computing models, paradigms, and protocols
- Object models and information modeling

– *Telecommunications Technologies:*
- Languages for service specification and implementation
- OSs & compilation techniques for services provisioning
- Issues of performance, transparency, portability
- Robustness and security
- Network and computing technology enablers

– *Computing and Communications Services and Applications:*
- Multi-party cooperation and groupware applications
- Mobile computing, universal personal telecommunications
- Video on demand servers and applications
- World-Wide Web servers, browsers, and features
- Multimedia systems and applications/Conferencing/CSCW
- Broadband Virtual Private Networking
- Network publishing, On-line Newspapers
- Tele-education, Remote instructional systems
- Electronic malls/Home shopping
- Telemedicine/Medical imaging

In 1996, the TINA organizing committee intends to formulate a program that reflects a broad spectrum of research. This will include research specifically targeted to telecommunications applications, but also distributed computing research of a more general nature, provided that the relevance to telecommunications systems is clear. Papers reporting on leading-edge products and practices are also welcome provided that significant scientific content is demonstrated. The breadth of the program will be expressed through the paper selection process, the identification of appropriate keynote speakers, and through demonstration and "work in progress" sessions which will highlight practical work of important current interest. As the conference evolves, it is positioned to emerge as the preeminent forum for the presentation of major cross-cutting results in the field. TINA '96 is committed to a very stringent selection standard. Accepted papers will be expected to display a maturity of methodology, significant results, and to adhere to high literary standards. Papers should not have been published previously. In many cases, authors will be asked to revise their submissions to respond to concerns or recommendations from the program committee. At the same time, the conference seeks to commend exceptional papers, and will do so by selecting a small number of submissions to receive "outstanding paper" awards.

Demonstrations

Demonstrations of advanced communications and distributed computing technologies, products, and prototypes of advanced research systems, are sought for TINA '96. These should implement and illustrate some elements of the emerging telecommunications or distributed computing architecture. Proposals for demonstrations, including any local support requirements, are to be submitted to the correspondence address as early as possible.

Submissions

Authors are requested to provide 2 printed copies of each submission, as well as an HTTP address for an electronic copy if possible. The address for submitted papers is:

TINA '96 Conference Submissions
EURESCOM
Schloss-Wolfsbrunnenweg 35
D-69118 Heidelberg
GERMANY
Tel.: +49-6221 989-100
Fax: +49-6221 989-209
E-mail: `TINA96@eurescom.de`

Submitted papers should not exceed 12 printed pages (approximately 5000 words), when formatted in a standard 10-point font on a standard A4 or 8.5 x 11 page with normal line spacing and margin sizes. This length restriction includes all figures and illustrations, references, title page, and other material that will become part of the finished document. The names of the authors and their institutional affiliation should not appear on the paper itself. However, an information sheet giving this information, as well as full contact information for the contact author, should be included with each submission.

Venue

TINA '96 is sponsored by the Telecommunications Information Network Architecture Consortium and by EURESCOM, a research consortium of European Telecommunications Companies. Heidelberg is a small, historic city located on the banks of Germany's lovely Necktar river, and is easily reached by train or air. Renowned for its stunning location, outstanding local foods, and exceptional wines, the city promises both a stimulating setting for the creative discourse that characterizes TINA conferences and a unique recreational and sightseeing opportunity for attendees.

continued on next page

Announcement and Call for Papers *(continued)*

A banquet in the historic castle overlooking the city is planned as part of the social program, and the conference itself will be sited in the Stadthall in the center of the old part of the city. Tours of the region can be arranged with several agencies that operate from the city.

Tutorials and workshop

The conference will be preceded (on Monday, September 2), by a day of Tutorials and a workshop organized by *reTINA,* the real-time special interest group of the TINA consortium. Information on the tutorials and workshop, including fees for tutorial attendees, will be published with the final conference program. The Program Committee Chairman for the reTINA workshop is Dr. Thomas Jell, Siemens-AG Corporate Research, e-mail: `Thomas.Jell@zfe.siemens.de.`

Several TINA sponsors will be inviting attendees to visit their German facilities on Friday September 6. Details of visits and information on participation will be available prior to the conference.

Important dates

Deadline for paper submissions:	March 1, 1996
Notifications of acceptance:	May 1, 1996
Camera ready papers due:	July 1, 1996

The Internet Traffic Archive

The *Internet Traffic Archive* (ITA) is a moderated repository to support widespread access to traces of Internet network traffic. The traces can be used to study network dynamics, usage characteristics, and growth patterns, as well as providing the grist for trace-driven simulations. The archive is also open to programs for reducing raw trace data to more manageable forms, for generating synthetic traces, and for analyzing traces. The archive is available on the Web as:

`http://town.hall.org/Archives/pub/ITA/`

Traces

There you will find a description of the archive, its associated mailing lists, the moderation policy and submission guidelines, and the initial contents of the archive, which includes the following traces:

BC: 4 million-packet traces of LAN and WAN traffic seen on an Ethernet (includes some traces from the Bellcore "self-similarity" study).

DEC-PKT: 4 hour-long traces of all wide-area packets.

LBL-TCP-3: 2 hours of wide-area TCP packets.

LBL-PKT: 2 hour-long traces of all wide-area packets.

LBL-CONN-7: 30 days of wide-area TCP connections.

...and the following software:

sanitize: a collection of shell scripts for "sanitizing" *tcpdump* trace files to address privacy and security concerns.

tcp-reduce: a collection of shell scripts for reducing a *tcpdump* trace file to a summary of the corresponding TCP connections.

fft-fgn: an S program for synthesizing self-similar processes.

Credits

The ITA was put together by Peter Danzig (USC), Jeff Mogul (DEC-WRL), Vern Paxson (LBNL), and Mike Schwartz (UCol Boulder), and was made possible by Carl Malamud and the Internet Multicasting Service giving it a home.

—Vern Paxson
Lawrence Berkeley National Laboratory
`vern@ee.lbl.gov`

Future NetWorld+Interop Dates and Locations

NetWorld+Interop 96	Las Vegas, NV	April 1–5, 1996
NetWorld+Interop 96	Frankfurt, Germany	June 10–14, 1996
NetWorld+Interop 96	Tokyo, Japan	July 22–26, 1996
NetWorld+Interop 96	Atlanta, GA	September 16–20, 1996
NetWorld+Interop 96	Paris, France	October 7–11, 1996
NetWorld+Interop 96	London, England	Oct. 28–Nov. 1, 1996
NetWorld+Interop 96	Sydney, Australia	November 25–29, 1996
NetWorld+Interop 97	Singapore	April 7–11, 1997

All dates are subject to change.

More information

Call 1-800-INTEROP or +1-415-578-6900 for more information. Or send e-mail to `info@interop.com` or fax to +1-415-525-0194. For the latest information about NetWorld+Interop including *N+I Online!* as well as other SOFTBANK produced events, check our home page at `http://www.interop.com`

NetWorld+Interop is produced by SOFTBANK Exposition and Conference Company, 303 Vintage Park Drive, Foster City, California 94404–1138, USA.

Write to *ConneXions*!

We'd love to hear your comments, suggestions and questions about anything you read in *ConneXions*. Our editorial address is given below. Use it for letters to the Editor, requests for the index of back issues, questions about particular articles etc.:

ConneXions—The Interoperability Report
303 Vintage Park Drive
Suite 201
Foster City
California 94404–1138
USA
Phone: +1 415-578-6900 or 1-800-INTEROP (Toll-free in the USA)
Fax: +1 415-525-0194
E-mail: `connexions@interop.com`
URL: `http://www.interop.com`

Subscription information

For questions about your subscription please call our customer service hotline: 1-800-575-5717 or +1 610-892-1959 outside the USA. This is the number for our new subscription agency, Seybold Publications. Their fax number is +1 610-565-1858. The mailing address for subscription payments is: P.O. Box 976, Media, PA 19063–0976.

CONNEXIONS
303 Vintage Park Drive
Suite 201
Foster City, CA 94404-1138
Phone: 415-578-6900
FAX: 415-525-0194

ADDRESS CORRECTION
REQUESTED

FIRST CLASS MAIL
U.S. POSTAGE
PAID
SAN JOSE, CA
PERMIT NO. 1

CONNEXIONS

EDITOR and PUBLISHER — Ole J. Jacobsen

Printed on recycled paper